US-PAT-NO: 5050213

DOCUMENT-IDENTIFIER: US 5050213 A

TITLE: Database usage metering and protection system and method

---------- KWIC ---------

Abstract Text - ABTX (1):
A "return on investment" digital database usage metering, billing, and security system includes a hardware device which is plugged into a computer system bus (or into a serial or other functionally adequate connector) and a software program system resident in the hardware device. One or more **data bases are encrypted** and stored on a non-volatile mass storage device (e.g., an optical disk). A tamper-proof decrypting device and associated controller decrypts selected portions of the stored database and measures the quantity of information which is decrypted. This measured quantity information is communicated to a remote centralized billing facility and used to charge the user a fee based on database usage. A system may include a "self-destruct" feature which disables system operation upon occurence of a predetermined event unless the user implements an "antidote"--instructions for implementing the antidote being given to him by the database owner only if the user pays his bill. Absolute database security and billing based on database usage are thus provided in a system environment wherein all database access tasks are performed at the user's site. Moreover, a free market competitive environment is supported because literary property royalities can be

calculated based on
actual use.

Brief Summary Text - BSTX (32):
In accordance with one important feature of the present invention, a storage
medium stores the database in **encrypted form, and also stores index information**
which correlates portions of the encrypted database with index keys. The index
information may itself be encrypted if desired. A host digital signal
processor operatively connected to the storage medium is preprogramed so as to
generate a database access request, read the index information from the storage
medium, identify (in accordance with the index **information) the portions of the**
encrypted database which satisfy the access request, and read the identified
encrypted database portions from the storage medium.

Detailed Description Text - DETX (8):
The database is preferably "preprocessed" and then stored onto medium 100.
The type of preprocessing performed depends upon the database and the
application, but typically includes creating an encrypted rendition of the
database and loading the encrypted rendition onto medium 100. One or more of
the many sophisticated conventional **data encryption** schemes which presently
exist can be used for encrypting the database. Preprocessing preferably also
includes generating an index to the database and storing the index together
with the encrypted version of the database on the storage medium 100. The
index may or may not be encrypted.

Detailed Description Text - DETX (10):
FIG. 2 shows one exemplary scheme for storing database information on medium

100. The information stored on medium 100 includes an index portion 102 and an encrypted database portion 104. Database portion 104 includes a plurality of predefined quantities, or "blocks", 106 of digital data. Each block 106 includes three information "fields": an index key field 108a; an **encrypted database information** field 108b; and a decryption key/error-checking field 108c.

Detailed Description Text - DETX (11):

Index portion 102, which may be **encrypted, provides information** used to translate a database access request into the addresses of one or more blocks 106. The contents of index portion 102 depends on the type of database stored on medium 100 and the type of operations which are to be performed on the database. For example, if word or string searching is to be provided, index portion 102 may include a list of all of the words contained in the database and the blocks 106 in which the listed words appear. Index portion 102 may alternately (or also) include a "table of contents" of the database and a designation of the blocks 106 covering each entry in the table. Other ways to index a database are known, and the present invention is not limited to any particular indexing scheme.

Detailed Description Text - DETX (13):

Encrypted database information fields 108b contains predetermined portions of the encrypted database. The size of these portions may be determined by the particular hardware and/or encryption techniques used, and is preferably (but need not be) fixed. If the nature of the database permits, logically-related information should be stored in the same blocks 106 (i.e.,

the database should
be presorted and hierarchically organized) to reduce the number of accesses of
storage medium 100 required to respond to a single user request. Techniques
for organizing databases are known to those skilled in the art of information
retrieval and database design and management.

Detailed Description Text - DETX (14):
Decryption key/error-checking field 108c performs two functions in the
preferred embodiment. First, it provides conventional error checking (e.g. CRC
or parity) information useful for detecting information reading errors.
Secondly, the field may provide information needed by sophisticated data
decryption schemes to decrypt the information stored in associated field 108b.
In many data decryption schemes, a decryption key word (which may itself be
encrypted) carried with the encrypted data is used in conjunction with an
additional data decryption key generated by the data decrypting device to
decrypt the data. Field 108c may or may not be required depending upon the
error checking and decryption schemes employed.

Detailed Description Text - DETX (16):
When a user requests information from the database stored on storage medium
100, the computer program resident on computer 200 controls hardware of the
computer to read the index information 102 stored on medium 100 in order to
ascertain which database blocks 106 contain information specified by the user
request. The computer program then controls host computer 200 to load one or
more blocks 106 of the stored database information into the host computer
memory. The host computer 200 then, under software control, strips off the

contents of encrypted fields 108b from the blocks of information now resident in its memory (along with some or all of the contents of decryption key/CRC field 108c) and sends some or all of this information to the decoder/biller block 300 for processing.

Detailed Description Text - DETX (18):

If index portion 102 is encrypted, it must be decrypted before a user can make selections from it or otherwise use it to locate blocks 106. Decryption of index portion 102 should be performed in a secure environment (such as in decoder/biller block 300, or in a dedicated "browsing workstation" to be discussed in connection with FIG. 5). Alternatively, decoder**/biller** block 300 may temporarily provide host computer 200 with the decryption key **information** needed to decrypt index portion 102 (the index portion may be encrypted using an encryption technique which is **different from the one used** to encrypt database portion 104), and the host computer can decrypt sections of the index portion as needed by the user.

Detailed Description Text - DETX (21):

Decoder/biller block 300 measures the amount and/or type of information sent to it for decryption and stores information indicating database usage over time from such measured amounts. Decoder**/biller** block 300 stores all necessary **billing** and usage **information** in a protected, non-volatile memory device (or in a protected, non-volatile storage facility within the host computer 200) for **later retrieval and use in calculating database usage fees**.

Detailed Description Text - DETX (22):

Because the database information read from medium 100 is

useless unless it
is first decrypted, and decoder/biller block 300 is the only portion of system
10 which is capable of decrypting the **encrypted database information,** the
decoder/biller block can accurately meter the amount and nature of data
accessed from the stored database e.g., by counting the number of blocks 106
which are encrypted, determining the group of logically related information
("property") stored on medium 100 which is logically associated with the data
being decrypted, and/or determining other convenient parameters indicating the
quantity and/or identity of data which is decrypted]. Decoder/biller block 300
decrypts the information sent to it, and returns the decrypted information to
host computer 200 for display, storage, printing, telecommunications, or the
like (or otherwise makes the decrypted information available to the user).

Detailed Description Text - DETX (23):

FIG. 3 is a more detailed schematic diagram of the decoder/biller block 300
shown in FIG. 1. Block 300 includes the following: a tamper-proof mechanism
302; a data connector 304 for connection to the host computer 200; a data
connector 306 for connection to an off-site service company; host computer
interface **logic 308; database decryption logic** 310; interface **logic 312; a**
non-volatile memory 314; decoder control logic 316; and a real-time
clock/calendar 318.

Detailed Description Text - DETX (25):

Another safeguard against tampering can be provided by implementing one of
more of functional blocks 308-318 in the form of a custom integrated circuit.
Such custom integrated circuits are not easily reproducible

by an unauthorized
person, nor could functional equivalents be designed ("black-boxed") so long as
the techniques used to encrypt and decrypt the database are sophisticated.
This level of **data encryption** sophistication is well within present technology.

Detailed Description Text - DETX (29):
Decoder control logic 316 preferably includes a conventional microprocessor
pre-programmed with a predetermined control computer program, but might be
implemented in other ways (e.g., as a discrete digital logic sequential state
machine). Decoder control logic 316 controls all of the functions of
decoder/biller block 300 in the preferred embodiment. Decoder control **logic**
316 also monitors database usage, produces digital data indicating the amount
of such usage, and stores this data in non-volatile memory 314 for later
retrieval (e.g., by a service company or the database owner).

Detailed Description Text - DETX (32):
Database decryption logic 310 takes input digital data signals provided to
it by decoder control logic 316 (these signals representing **encrypted digital**
data read by host computer 200 from storage medium 100 and passed to the
decoder control logic via connector 304 and interface logic 308), decrypts
these digital data signals using a predefined decryption algorithm, and outputs
decrypted data signals to the decoder control logic for display, printing, and
the like. One or several different predefined decryption algorithms can be
stored in (or hardwired within) decryption logic 310, and additional decryption
algorithms can be downloaded into the decoder/biller block 300 as needed or

required via interface logic 312.

Detailed Description Text - DETX (33):
Many conventional methods of **encrypting/decrypting data** are known, spanning from simple lookup tables to complex mathematical algorithms. The method of **data encryption**/decryption used depends on the amount of extra computer processing overhead and data storage space that the application will allow. It is not uncommon for substantial overhead to be needed to handle **encrypted data**.

Detailed Description Text - DETX (42):
System 10 may require the user to input identification and/or password information along with his access request (block 404). System 10 checks the authority of the user to access the database by transmitting the inputted ID/password information to decoder/biller block 300 for comparison with a list of authorized IDs/passwords stored in memory 314 (block 410). If decoder/biller block decoder control **logic 316 denies authorization to continue with database** access (because the inputted user information is incorrect, because the access request cannot be performed at the current time/date. etc.) (block 412). the decoder/biller block refuses to decrypt any data sent to it (block 414)--and may cease communicating with the host computer 200, and/or simply ignore any **encrypted information** the host computer sends it. While **encrypted database information** is already present in the memory of host computer 200, this **encrypted information** is incoherent and cannot be used for any useful purpose.

Detailed Description Text - DETX (43):

On the other hand, if decoder control logic 316 of decoder/biller 300 grants authority to proceed (block 412), the decoder control logic begins a "billing cycle", and stores information logging the billing cycle into non-volatile memory 314 (block 416). The information stored in memory 314 may include: (a) the name of the database file being accessed; (b) the section of the database being accessed (name, "property designation", file name, or other identification information); (c) the identification of the user accessing the database; and (d) the date and time the database access begins.

Detailed Description Text - DETX (44):

The information stored in non-volatile memory 314 may thus be used to create an "audit trail" which tracks different users (or groups of users) and their database usages. Special use passwords may be required to access selected databases, and actual use of all databases may be verified later from the information stored in memory 314. Such stored information is extremely valuable not only to help detect unmonitored database use, but also to allow detailed bills to be generated and to help determine which users among multiple users are responsible for generating usage charges. Such a detailed audit trail can be used to allow publishers and users to determine the detailed activities of users. This information can be used by users to determine what they are being charged for. The audit trail information can also be used by publishers and property owners to conduct marketing surveys--providing more detailed information about user demographics and information use than is presently available.

Detailed Description Text - DETX (45):
In addition, it may be desirable to code storage medium 100 (or particular databases or files stored on the medium) with unique (e.g., randomly-generated) user passwords by embedding secret password information in the database information. Non-volatile memory 314 can store information which matches the code associated with the particular copy of the storage medium licensed to a particular user. This coded **information can be encrypted,** and coding schemes and/or coded information may be changed periodically. Different users can be assigned different codes to prevent users from exchanging or sharing storage media 100.

Detailed Description Text - DETX (47):
Decoder control logic 316 also is enabled at this time to begin (a) decrypting information sent to it by host computer 200 and (b) sending the decrypted information back to the host computer (block 418). Decoder control logic 316 meters the quantity and/or other usage parameters of data which is decrypted, and stores this usage information into non-volatile memory 314 along with the other billing information (block 420) (the decoder control **logic may store** quantity information directly into the memory, or may first convert it to billing information taking into account, for example, the cost of using the database file being accessed). This process continues until the user's request has been satisfied (as tested for by block 422).

Detailed Description Text - DETX (49):
The specific steps performed to decrypt data (block 418) depends on the

particular **data encryption**/decryption scheme used. Host computer 200 transmits **encrypted data** in predetermined quantities (e.g., fixed-length blocks) to interface logic 308 via connector 304 in the preferred embodiment. Interface logic 308 **communicates this encrypted data** to decoder control logic 316, which **communicates it to data encryption**/decryption logic 310. Logic 310 translates the **encrypted data** into intelligible information using a predetermined conventional decryption algorithm, and communicates the decrypted data back to decoder control logic 316. Decoder control logic 316 then communicates the decrypted data to host computer 200 via interface logic 308 and connector 304.

Detailed Description Text - DETX (51):

Decoder control **logic 316 meters database** usage (block 420) by, for example, measuring the amount of information which is decrypted (e.g., by counting the number of fixed-length blocks which are decrypted; determining the source documents the decrypted information is associated with; and measuring the time, date and/or duration of access of the decrypted information). Control logic 316 may also record other billing information, such as the length of the database file being opened. Control **logic** 316 may be arranged to recognize the names or other designations of subsections of the database being **accessed, allowing for different billing** rates depending on the type or supplier of the **information** (so that use of more expensive databases can be **billed** at higher rates).

Detailed Description Text - DETX (53):

After the user's access request has been satisfied (as tested for by block

422). the decoder control **logic stores,** into non-volatile memory 314, the time
the user finishes accessing the database. (block 424). The resident program
then allows the user to input another access request (using the same or
different database) (block 426). If the user does input another access
request, the steps of blocks 404-426 are performed again (with blocks 416, 420
and 424 causing an additional billing **log entry to be stored** in memory 314).

Detailed Description Text - DETX (54):
The information stored in memory 314 is periodically communicated to the
service company and used to bill the user for database usage. In one exemplary
embodiment, memory 314 is housed in a storage module 314a which is easily
separable from system 10. Periodically, the user disconnects memory module 314
from decoder/biller block 300, mails the module to the service company, and
installs an alternative replacement module (the "next" module) into system 10.
Decoder control logic 316 disables data decryption unless a module 314a is
connected to it (and perhaps also when the control **logic has determined the
non-volatile storage** area is nearly full).

Detailed Description Text - DETX (55):
In another embodiment, communications between decoder/biller block 300 and
the service company is periodically established for the purpose of downloading
the contents of memory 314 to the service company billing computer. If
connector 306 and programming interface logic 312 comprise a conventional
standard telephone connector and associated modem, such communications can be
established over standard telephone lines. The information stored in memory

314 is transmitted over the telephone line to the service company computer, and the service company computer then transmits commands which control decoder control **logic 316 to reset the memory**. In addition, the service company can establish communications with decoder/biller block 300 to monitor use of the databases stored on medium 100 (and detect misuse and unauthorized use). The service company may also control decoder/biller block 300 remotely (e.g., to disable it from operating if customer fails to pay his bill).

Detailed Description Text - DETX (74):

For example, although it may be undesirable to permit data type decryption key information to reside in the host computer permanently, the decryption key information can be temporarily provided by a protected memory device to the host computer. The host computer may then decrypt database information using the decryption key information, and destroy the key information after use. The host computer may decrypt database **information "on the fly" and not retain much encrypted or decrypted information** in memory at any one time to help prevent copying.

Detailed Description Text - DETX (75):

Although a dedicated hardware/software system typically provides the best assurance against tampering, techniques which may be implemented in software executing on a non-dedicated system may provide sufficient tamper resistance for some applications. For example, secure program control and usage information can be stored on a floppy disk which is accessed via the disk drive of a general-purpose non-dedicated personal computer. A non-volatile **memory**

and logic device connected to the personal computer may (in conjunction with
the secure program control software executing on the computer and/or a hardware
controller connected to the computer) control and monitor the position of the
read/write head of the disk drive, store the current head position in the
non-volatile memory, and supervise execution of the secure program control
software. Database usage information may be gathered by the program control
software and stored on the floppy disk. Any attempts to tamper with the floppy
disk which alters the last read/write head position may cause a warning message
to be **stored on the floppy disk in a database audit** trail section of the disk
(possibly along with cumulative messages indicating previous such occurrences)
and may also result in destruction and/or disablement of the secure program
control software.

Claims Text - CLTX (25):
at least one storage medium located at said user site and adapted to be
insertable into and physically removable from said housing by said user, said
at least one storage medium comprising an optical storage medium, said at least
one storage medium storing digitally encoded database **information that is, at
least in part, encrypted**;

Claims Text - CLTX (46):
a storage arrangement storing at least one database at a customer site, said
at least one database having at least one **encrypted part, and also storing
information** representing at least one database usage ceiling corresponding to
at least one portion of said at least one database;

Claims Text - CLTX (105):
at least one storage medium located at a customer site and storing database information on at least one removable, optical storage disc, with at least one part of said database **information being stored encrypted** form;

Claims Text - CLTX (113):
at least one optical disk having **encrypted digitally encoded database information** stored thereon;